REMARKS

Amendments to claims 1, 11, and 23 are for the purpose of clarifying what Applicants regard as the invention. Amendments to claims 4, 6, 14, and 16 are to change claim dependencies. No new matter has been added.

I. CLAIM OBJECTION

Claims 4-8 and 14-18 stand objected to as being of improper dependent form. Claims 4, 6, 14, and 16 have been amended to change claim dependencies. Applicants respectfully submit that claims 4, 6, 14, and 16, and their respective dependent claims, are in condition for allowance.

II. DOUBLE PATENTING REJECTION

Claims 1-23 stand rejected under the judicially created doctrine of obviousness-type double patenting over claims 1-37 of U.S. Patent No. 6,112,197. Attached herewith is a Terminal Disclaimer disclaiming that portion of the term of any patent granted on the present application extending beyond the term of United States Patent No. 6,112,197. It is believed that the Terminal Disclaimer overcomes the double patenting rejection.

III. CLAIM REJECTIONS UNDER U.S.C. § 102

Claims 1, 11, and 23 stand rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 6,178,461 (Chan). Applicants respectfully notes that in order to sustain a rejection under §102, each element in the rejected claim must be found, either expressly or inherently, in the cited reference.

Claims 1, 11, and 23 each recites determining the existence of data redundancies in prefetch data, obtaining a reduced set of prefetch data based at least in part on the determined existence of data redundancies, and transmitting the reduced set of prefetch data from a server to a client, wherein the reduced set comprising a smaller memory footprint than the prefetch data. Chan does not disclose or suggest such limitations. In the Office Action, column 2, lines 10-60, and column 3, lines 3-20, and column 11, lines 29-47 were cited as the basis of the 102 rejection. However, column 2, lines 10-60 actually disclose:

> Another technique, known as "prefetching", tries to predict, fetch and store a response before it is needed. In both caching and prefetching, the stored objects must be transferred at least once. In some cases, this transfer can be optimized by first transforming or compressing the objects to a smaller size. Prefetching, like caching, is based on fetching in advance the exact object that will be needed in the future. The utility of prefetching was studied by Venkata N. Padmanabhan and Jeffery C. Mogul, "Using Predictive Prefetching to Improve World Wide Web Latency", Computer Communication Review, ACM, July 1996, using a statistical algorithm described by James Griggioen and Randy Appleton in "The Design, Implementation, and Evaluation of a Predictive Caching File System", Technical Report CS-264-96, Department of Computer Science, University of Kentucky, Lexington, Ky., June 1996.
>
> *Compression can be achieved by the use of differential transfer to transmit only changes between current and past information.* (Emphasis Added) Usually, only two objects of the same URL are considered at a time. Some of the differencing algorithms used are UNIX diff and vdelta, as described by J. Hunt, K. P. Vo, and W. Tichy, "An Empirical Study of Delta Algorithm", IEEE Software Config. and Maint. Workshop, March 1996; and J. Hunt, K. P. Vo, and W. Tichy, "An Empirical Study of Delta Algorithms", IEEE Software Config. and Maint. Workshop, March 1996; In a paper by Gaurav Banga, Fred Douglis, and Michael Rabinovich, "Optimistic Deltas for WWW Latency Reduction", USENIX, 1997, the difference between two versions of the same page was computed in order to reduce transfer size. Dynamic pages including output of CGI scripts with different parameters, were also considered for differencing. The issue of what objects should be used in differencing has been mentioned in the literature, but is at present considered to be an open question. The benefits of delta coding was also studied by Jeffery C. Mogul, Fred Douglis, Anja Feldmann, and Balachander Krishnamurthy, "Potential Benefits of Delta Encoding and Data Compression for Http", Proceedings of the ACM SIGCOMM, pages 181-194, 1997.
>
> The limits of latency reduction obtainable from caching and prefetching, based on search for objects with same the URL, was studied by Thomas M. Kroeger, Darrell D. E. Long, and Jeffrey C. Mogul, "Exploring the Bounds of Web Latency Reduction from Caching and

> Prefetching", USENIX Symposium on Internet Technologies and Systems, December 1997. They have found that caching and prefetching can reduce latency by at best 26% and 57%, respectively. Accordingly, there is a significant need for an improved latency reduction technique.

As such, the cited passage discloses compressing data by using differential transfer to transmit only changes between current and past information, and does not disclose or suggest determining the existence of data redundancies in prefetch data, obtaining a reduced set of prefetch data based at least in part on the determined existence of data redundancies, and transmitting the reduced set of prefetch data from a server to a client, as recited in claims 1, 11, and 23.

Also, column 3, lines 3-20 actually disclose:

> More specifically, when a user requests an object, and that object is not already in the client's local cache, similar objects in the local cache are identified, and a request is sent to the level-1 proxy server to retrieve the desired object, using a set of stored objects that are similar to the requested object. If the requested object is not in that server's cache, it is fetched from the origin server. However, instead of sending the actual object over the last hop between the level-1 proxy server to the client, the object is encoded using as reference objects the similar objects that are already available in the cache. The more "similar" the reference objects are to the requested object and the more such "similar" reference objects are available in the client's possession, the smaller is the resulting transfer. The encoded information received by the client processor is then decoded in conjunction with the set of similar objects in the local cache that were previously identified, and the decoded information is presented to the browser application.

As such, the cited passage discloses retrieving requested object by identifying similar object in a local cache. There is nothing in the cited passage that discloses or suggests determining the existence of data redundancies in prefetch data, obtaining a reduced set of prefetch data based at least in part on the determined existence of data redundancies, and transmitting the reduced set of prefetch data from a server to a client, as recited in claims 1, 11, and 23.

Also, column 11, lines 29-47 actually disclose:

> Another useful contribution of the present invention is that it nicely ties together compression, caching and prefetching. It relates compression and caching by providing a way to leverage the cached objects to improve compression. For prefetching, one can potentially use the idea of compaction to hedge against prefetching the "wrong" objects by prefetching instead a set of objects that are most likely to be "similar" to the objects that will be needed in the future. This relaxes the selection criteria for prefetching and significantly alleviates the penalty of incorrect prefetching. Our compaction technique also appears to be relatively immune from the changing nature of Web content. One of the main trends in Web content is the increasing use of CSS, DHTML, and even XML. All of these, especially XML which employs application-specific tags, are well suited for compaction. Specifically, they all try to introduce consistency and uniformity across Web documents. This translates into useful "similarity" that can be exploited using compaction.

As such, the cited passage discloses using compaction technique to prefetch a set of objects. There is nothing in the cited passage that discloses or suggests determining the existence of data redundancies in prefetch data, obtaining a reduced set of prefetch data based at least in part on the determined existence of data redundancies, and transmitting the reduced set of prefetch data from a server to a client, as recited in claims 1, 11, and 23.

For at least the foregoing reasons, claims 1, 11, and 23, and their respective dependent claims, are believed allowable over Chan.

IV. CLAIM REJECTIONS UNDER U.S.C. § 103

Claims 1, 2, 3, 9, 11, 12, 13, 19, and 22 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,098,064 (Pirolli) in view of U.S. Patent No. 5,600,316 (Moll). Applicants agree with the Examiner that Pirolli does not disclose or suggest determining redundancy of prefetch data. However, Moll fails to make up the deficiency present in Pirolli. According to the Office Action, column 12, lines 33-56, the abstract, and figures 8a-c of Moll disclose determining redundancy of prefetch data. However, the cited passages actually disclose:

> Repetitive data and non-repetitive data, including periods of no information, is encoded prior to transmission or storage in digital form to minimize the memory space required for storage or the time required to transmission. Repetition, partial repetition, and near repetition is encoded in a form indicating the occurrence of repetition, its characteristics and its duration. The existence and size of repeated patterns in the data is dynamically determined. When repetition is detected, non-repetitive data is inserted into the data stream and repeated data is removed from the data stream. To this non-repeated data in the data stream are added a coded repeated pattern sample, an identification preamble signal, an instruction signal for decoding purposes, a period count signal, a mask signal, and a repeat count signal. All necessary data elements are combined and assembled to produce compressed data. A receiver utilizes these coded and uncoded data elements to regenerate complete original data.
>
> The basis for data compression is to remove the repeated bits because they need not be stored, but rather regenerated at a later time.
>
> FIG. 8a shows five 4-bit characters where the first bit of each character is a one, and the third bit of each character is a zero. This example of the invention removes the 1st and 3rd bits of each character and sends the remaining bits, in this case as 2 bit characters. To designate which bits to remove (and later regenerate), the control data contains a mask which has the 51st and 3rd bits set to one (remove) and the remaining bits reset to zeros (retain). The first 4 bit character of the data involved in repetition is transmitted as the pattern, such as pattern 457 in FIG. 4. The pattern shows the state (1's or 0's) of the repeated bits for later use when regenerating the data.
>
> FIG. 8b shows a similar case using a CVSD coded audio frequency as an example. FIG. 8c shows a similar case using ASCII coded data. FIG. 8d shows the technique of removing the repeated bits before storage or transmission using the data of FIG. 8a. The mask bits are in an end around mask shift register 890 moving at the same clock rate as the shifting in data register 114. The mask register bits cause the data register bits to be either thrown away via AND gate 887 or to be stored or transmitted via AND gate 889.

As such, the cited passages disclose compressing data by removing repeated data, and does not disclose or suggest determining redundancy of *prefetch* data, as recited in claims 1, 11, and 23. In addition, Applicants respectfully note that neither Pirolli, Moll, nor their combination discloses or suggests *obtaining a reduced set of prefetch data based at least in part on the determined existence of data redundancies*, and *transmitting a reduced set of prefetch data* from a server to a client, as recited in claims 1, 11, and 23. For at least the foregoing reasons, claims 1, 11, and 23, and their respective dependent claims, are believed allowable over Pirolli, Moll, and their combination.

Claims 1, 2, 3, 9, 11-13, 19, and 22 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,822,749 (Agarwal) in view of Moll. Applicants agree with the Examiner that Agarwal does not disclose or suggest determining redundancy of prefetch data. However, as similarly discussed previously, Moll fails to make up the deficiency present in Agarwal. For at least the foregoing reason, claims 1, 11, and 23, and their respective dependent claims, are believed allowable over Agarwal, Moll, and their combination.

V. INFORMATION DISCLOSURE STATEMENTS

Applicants submitted Information Disclosure Statements on November 25, 2003 and July 7, 2004, copies of which are attached hereto. However, we have not yet received confirmation that the references listed on the corresponding forms PTO/SB/08a have been initialed and considered. Applicants hereby respectfully request that the references on each form PTO/SB/08a be initialed and considered by the Examiner.

CONCLUSION

Based on the foregoing, all remaining claims are believed in condition for allowance. If the Examiner has any questions or comments regarding this amendment, please contact the undersigned at the number listed below.

The Commissioner is authorized to charge any fees due in connection with the filing of this document to Bingham McCutchen's Deposit Account No. **50-2518,** referencing billing number **7011022001**. The Commissioner is authorized to credit any overpayment or to charge any underpayment to Bingham McCutchen's Deposit Account No. **50-2518,** referencing billing number **7011022001**.

Respectfully submitted,

Bingham McCutchen LLP

Dated: 2/24/05

By:
Gerald Chan
Reg. No. 51,541

Bingham McCutchen LLP
Three Embarcadero Center
San Francisco, California 94111
Telephone: (650) 849-4960
Facsimile: (650) 849-4800